Hitting the Stride with SAP XI:

On Whether Exchange Infrastructure Is “All That”

By Dave Bernard, KimberWorks, Inc.

Editor’s Note: *It's the oldest trick in the book: when you want the best information, go right to the source. In the case of XI, we went directly to the original XI product development team. Dave Bernard, the author of this comprehensive new SAPtips white paper on XI, was part of that first SAP® NetWeaver™ XI product team and lived to tell. He has watched XI evolve from its early, awkward “baby steps” days to the current full-featured incarnation. But still, the question remains: Is XI all it's cracked up to be? In this exclusive white paper, Dave Bernard answers that question by putting the evolution of XI in the context of the overall development of the Enterprise Application Integration (EAI) space. Using helpful functionality charts, Dave explains how SAP sees XI as the next-generation hub of the NetWeaver product line. So is XI ready for prime time, or not? As always, the answer is relative: it depends on the customer. In the last section of this white paper, Dave describes several different approaches to XI adoption and talks about the skills needed to ensure a successful XI implementation. Through this honest inquiry, Dave is able to strike a balance between pointing out the challenges XI is facing and the obstacles it has yet to overcome. There's no way to get the final word on XI in one white paper, but you'll finish this piece with a deeper understanding of how XI fits into SAP's NetWeaver architecture.*

I. Introduction

Application interfaces are about the worst thing a company can pursue. They are brittle and prone to snapping upon application upgrades. They are hard to understand, harder to implement, and even harder to support. They tie up scarce funds otherwise allocated to real business solutions, yet they contribute not a business solution, only a technical mechanism. They are among the single most costly development expenditures during a project implementation, and eat up a similarly disproportionate slice of the ongoing maintenance and support budget.

Yet, whether heterogeneous implementations arose spontaneously over time, or through a dogged pursuit of a best of breed strategy, those brooding and sullen interfaces are a fact of life in most organizations. And even if we can stir the internal applications to sing in harmony, we still need to automate interaction with external customers and suppliers.

Add to the challenge a quiet impending revolution by integration products. Rising above the mundane productivity benefit of integrating business processes, vendors hold out the tantalizing possibility of increasing market responsiveness and general adaptability to change. While we were busy thinking of integration products as merely a means to better coordinate the distribution of business processes, the products themselves pulled a judo move by offering to orchestrate a greater diversity of business applications, and to a brand new level, through approaches such as Service Oriented Architectures (and their offspring).

In Figure 1, the bottom Application Layer depicts the world of traditional interfaces – combinations of FTP, BDC, and shell scripts, for example. Moving up, the Integration Hub brokers flows across the applications, allowing many-to-many communication and simplified incremental application participation. This layer also permits an abstracted Service Layer of cataloged and published Web services, leveraging underlying functionality that may cross multiple applications. At the top, flexible xApps, or Composite Applications, can then selectively use those available services to create lightweight specialized applications (instead of old school custom development). The key to all this is the Integration Hub in the middle, without which there is no real application abstraction or data abstraction.

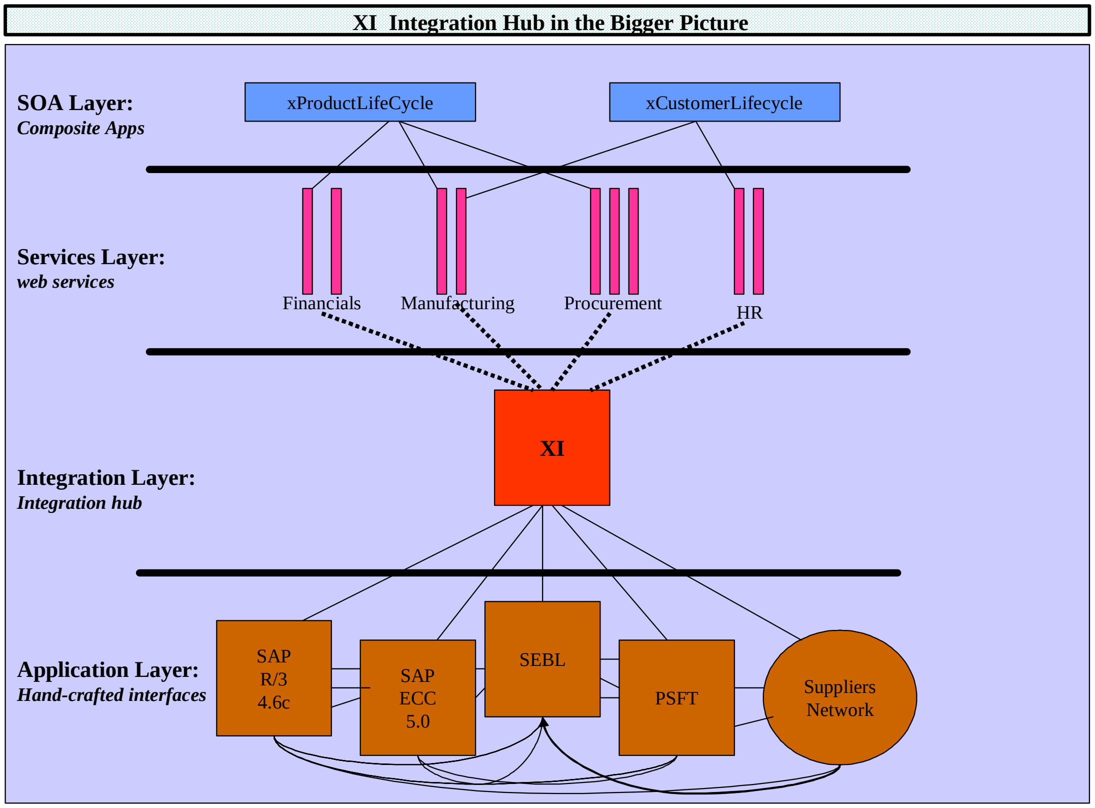


Figure 1: XI Integration Hub in the Bigger Picture

So before we can complete the wonderful SOA castle on the hill, we need to install the plumbing. SAP has promoted Exchange Infrastructure (XI) as just the product to serve as the Integration Hub. Yet XI is also criticized as being just a “me-too” product with limited functionality, late to the market, unlovely, and offered by a vendor that should stick to peddling business software.

It's time for a closer look at XI and its relation to the integration market. No longer a newly released product allowed the rookie's forbearance; we should examine it in the context of today's integration market in general, as well as in its own right. And if, indeed, integration products from ERP vendors are unavoidable, and if agile enterprises are desirable, we need to define evaluation criteria and adoption approaches for such enabling products.

I have had the opportunity to look “under the hood” of XI for a number of years now, initially while serving as part of the original NetWeaver XI practice at SAP America. In this white paper, I’ll take a close look at the current incarnation of XI, and whether it delivers on its marketing literature or not. Like most new SAP products, XI is less of a slam dunk and more of a work-in-progress. By examining the context that caused SAP to develop XI, this white paper will give you a better handle on how XI can help your SAP implementation and give you a framework for evaluating its pros and cons for your SAP environment.

II. The Integration Solution Landscape (and the Inevitability of XI)

The Integration Problem: Integration is Not Getting Simpler

We somehow find time to read trade journals and go to conferences. We give a nod to the concept and theoretical value of common integration patterns and architected service suites. We understand the potential of Web services, while denying the lingering unease arising from six years of reading about how companies are starting to use them, knowing that real mainstream technologies seldom merit hyperbolic press. Maybe what really disturbs us is that these solutions don't really seem to address the integration problem; quite the opposite. How do I respond when my boss asks why we have to get so much more complicated before we can get more simple?

Yet if we believe the surveys, integration continues to rank as one of the top five priorities of CIOs. We have no reason to believe that change won't continue, and have every incentive to address this very high priority. Our real challenge is to determine the best means of integration.

Application vendors themselves do not necessarily make things easier. In the mid-90s, the software shibboleth was the integrated ERP suite, combining in a single product accounting, manufacturing, and logistics; everything a company would want. Still, functionality gaps were soon identified and addressed by smaller software vendors. In an effort to stymie this growing competition, ERP vendors began moving to a dis-integrated approach, offering new customer relationship management (CRM), supply chain management (SCM), and electronic procurement products. These seemed to somehow have a similar, but vexingly different lineage from their ERP forbears, even when all were from the same vendor.

To address this apparent *dis*integration, independent Enterprise Application Integration (EAI) and Business-to-Business (B2B) vendors jumped on the opportunity in the late 90s to address The Integration Problem. But, in a similar effort to meet growing specialized competition (and themselves too frequently having limited start up funds), integration and technology vendors often turned to acquisitions, partnerships, and consolidations. Such vendors suddenly found themselves saddled with integrating their own integration products, until even today they may still be challenged in integrating their B2B, EAI, BAM, BPM, and Web service management. That's not to mention segmenting (and confusing customers) across an array of products with overlapping functionalities. There's no arguing that these vendors have been busy giving at least their front-end tools a similar look and feel, while perhaps hiding underlying complexities under the banner of a single brand. The upshot is still that in an effort to address changing market requirements, integration products themselves have too often become part of the integration problem.

Nevertheless, these vendors have been around for a long time (in software company years), have established market recognition, and productive, satisfied customer bases. Along comes SAP with XI.

Before we haze this rookie, consider SAP's track record in entering new markets: In 1996 SAP partnered with i2 and Manugistics to offer supply chain management solutions; and after later entering that space itself, SAP simply blew past them. In 1999 SAP's SRM (nee "B2B") product lagged behind Ariba and CommerceOne in almost every way imaginable. Not too long ago, CRM was a major gap in the SAP product line, challenged by a strong field of major independent CRM software vendors (R.I.P. all!). SAP came relatively late to the business intelligence and the portals space. In those spaces, SAP quickly became a major player. And although during the late 90s SAP was investing venture capital in at least one fledgling EAI vendor, and partnering for B2B solutions with another, it only now enters the integration arena, its path littered with the ruins of former competitors. There is no reason to think this juggernaut has played out.

At the same time the early, entrepreneurial, integration vendors have continued to fade away as independent concerns. From pioneer B2B vendors such as NetFish and Extricity, to pioneer EAI vendors such as ActiveWorks, Neon, SeeBeyond/STC, CrossWorlds and Mercator, a natural consolidation and industry acquisition pattern has continued, until today the key remaining independent EAI/B2B players can be counted on a single hand. Of the remaining independent players such as Tibco, webMethods and Vitria, (whose share price in some cases remains only a percentage point or two of their peaks), it is still an open question as to who is next, with the squeeze coming from both system vendors (e.g., Websphere, BEA) and application vendors (Oracle, SAP).

Ignoring for a minute the relative maturity, flexibility, or power of this or that integration product, long-term viability and support from the vendor have to be primary concerns to any customer. SAP at least has shown the legs to go the distance.

Even beyond the advantage of a strong software revenue number and well-funded product development, other indicators point to advantages for SAP's fledgling integration product.

A. Common Licensing
Although arrangements may vary (depending upon individual contracts), SAP has, by and large, bundled the free use of XI as part of the NetWeaver 04 license; as long as XI is used SAP to SAP. This means not only are organizations so licensed free to use XI to develop new integrations, it also, very importantly, means that XI maintenance fees are non-incremental. Here is a low-impact incentive to begin using XI immediately, whether for a proof of concept or for production-bound integrations. This provides an almost “why not” case for XI acceptance by SAP customers, as long as they keep in mind that non-SAP end points will entail volume-based fees, and non-technical adapters (see below) will be licensed separately.

I have more than one client that is either seriously planning, or actively rolling out XI, in an effort to specifically eliminate the very tidy annual maintenance costs of their existing integration product. These customers anticipate straight payback for the XI deployment within a relatively short time.

B. Built-In Application Support
SAP product development has already announced baked-in support for XI by new business apps. As these new product modules are introduced, SAP is all but mandating the use of XI for interfaces. Supplier Self Service, Event Manager, Master Data Management, and some aspects of the Business Warehouse are some examples. The SAP ABAP stubs (which SAP calls proxies) to support XI communication are provided as part of these products, as are the message definitions and structures for the SAP app-side of the XI scenarios.

And even for more mature SAP products, XI is quickly becoming the integration product of choice. For example, to send Purchase Orders from Enterprise Business Professional to suppliers, predefined proxies and XI content are also freely downloadable to licensed customers, allowing quick adoption of XI in what was once Business Connector territory.

Thus, there is a real question as to whether new BAPIs, RFCs, or IDOC processing will ever be provided for new applications (not to say that existing ones will ever go away). Add to that, the announced direction that all existing applications will, in effect, be retrofitted to meet the SOA/Web service paradigm. Customers wishing to retain release parity over time will, as a matter of course, have strong motivation to consider XI.

C. Monitoring and Administration

A deterrent to the deployment of new products is the need for new skills to monitor and maintain them. A whole new integration product that is vital to the business requires the IT department to learn new skills, and be ready to monitor and fix yet another product. While most products offer monitoring, tracing, and recoverability, XI additionally offers some interaction with the traditional SAP admin tools, such as CCMS and the Alert Framework, with the ability to build off existing skills.

D. Where Is XI Today?

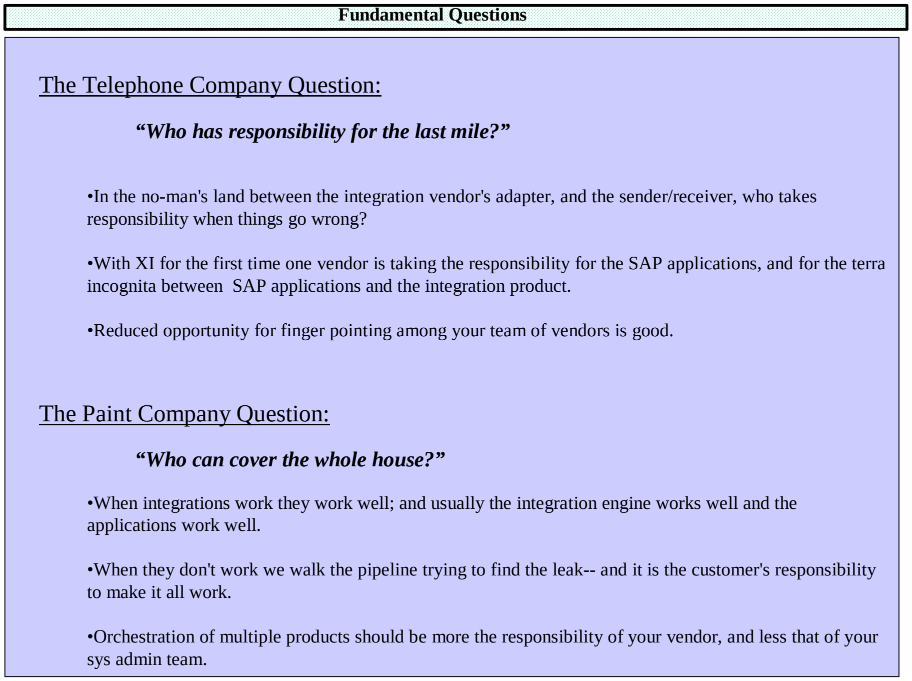
It's not always easy to gauge the extent of customer adoption of a new technology or product. Vendors will be glad to share numbers, as will industry analyst groups (for a price). XI has been around for over three years, but we'll hear on one hand that it's not ready for prime time, and on the other that it is being embraced. I like to use the Monster.com product popularity gauge. The results: through most of 2003, online XI help-wanted postings were mostly by SAP Labs or SAP America. The year 2004 began to see an upswing in postings, especially by the “Big n” and “Tier 1” consulting companies. These boys know their market well, and specialize in just-in-time “practice building” to meet a perceived demand. With 2005, following the release of XI 3.0, the postings have become increasingly diverse, from a mix of end-user companies offering full time openings, to a steady list of consulting engagements. The velocity of XI adoption is increasing, despite a market already in at least its seventh year of EAI presence. The pace of XI adoption may be slower than what SAP product management would like, but it is relentless.

III. Choosing an Integration Solution

In considering an integration product in general, a customer naturally should take development productivity claims with a grain of salt. Although reusability is at least a theoretical possibility, the time and effort to develop an SAP scenario (using even the best integration product) may often be disturbingly close to that using traditional means (such as custom Call Transaction, remote function call, and script-based ftp file manipulation).

The real payback for the product can more readily be observed over time and once in production, with highly reduced efforts to change, upgrade, administer, and recover from errors. The commonality of design patterns, together with the decoupling of the transport, transformation (and in some cases, business logic) from the application systems, can reward companies with a more streamlined and manageable environment. This enforced discipline translates into an annual cost savings and even a partial solution to the integration problem. It therefore frees up funds to be better deployed in delivering enhanced business services.

Vendors very naturally promote the strengths of their own solution models to the integration problem, be it a hub-and-spoke approach, a bus model, an event-driven infrastructure, or an enterprise service bus. Ignoring for the minute (regardless of the access metaphor), that the underlying technologies may be quite similar, I'd submit that most approaches could be equally successful. And that, without putting too fine a point on it, getting lost on technology ideology may delay us from asking more primary questions addressing issues that bedevil real life implementations (Figure 2).



Fundamental Questions

The Telephone Company Question:

“Who has responsibility for the last mile?”

- In the no-man's land between the integration vendor's adapter, and the sender/receiver, who takes responsibility when things go wrong?
- With XI for the first time one vendor is taking the responsibility for the SAP applications, and for the terra incognita between SAP applications and the integration product.
- Reduced opportunity for finger pointing among your team of vendors is good.

The Paint Company Question:

“Who can cover the whole house?”

- When integrations work they work well; and usually the integration engine works well and the applications work well.
- When they don't work we walk the pipeline trying to find the leak-- and it is the customer's responsibility to make it all work.
- Orchestration of multiple products should be more the responsibility of your vendor, and less that of your sys admin team.

Figure 2: Fundamental Questions

XI may not appeal to all businesses. Non-SAP implementers will be in the minority. But to SAP shops, it may not matter whether or not XI is the best of the best (assuming it is even possible to come up with a unanimous winner). All things being equal, XI just has to be good enough.

Which is not to say that XI will be a default choice to SAP customers, or that it is the best choice for any particular environment. Vendor or product selection is often an art, in that we need to base it on a complex list of requirements that we don't have well defined just yet. Most companies have some procedure in place for researching and selecting products, and integration products are not all that different.

A. Evaluation Criteria: Contrasting Features and Functions

There's no getting around the classic spreadsheet of requirements and features to be evaluated and compared across the short list of integration vendors. A starting point is to review current and future integration requirements, identify maintenance pain points, and spell out organizational standards and constraints. It may seem obvious, but a complete and up-to-date integration inventory is invaluable for gathering further information.

B. Bake-Off

Unlike ERP selection, the vendor bake-off is a classic technique long used to select integration products. The customer defines a limited number of specific representative scenarios to see which vendor can nail them quickest and easiest. When a good team of vendor staff is available to deliver the scenario, customer staff can uncover a remarkable amount of information in a short time.

However, the supporting framework may be complex to set up by the customer, and the results can simply be a result of which vendor coincidentally had the most experienced sales engineer available at the time, and hence not truly representative of the product's capability. Additionally, this approach tends to favor review of the development more than the runtime or support components. Because of the cost involved by customer and vendor, and the open question of the results, the pure bake-off does not seem as frequently used as in the past.

C. Reference Calls
Reference calls and visits remain one of the best ways to compare implementations. Special attention should be given to lessons learned, and suggestions as to what the reference customer would have done differently. With XI in the mix, it's important to target customers in production with the type and number of scenarios anticipated by the inquiring customer.

D. Vendor Viability
Sober actuarial calculations for vendor longevity should trump glamour. Long-term product maintenance and support may not actually be an immediate concern. There is usually a market to resell maintenance, even if the original vendor goes away. Still, future innovation would suffer. Implementing an integration product is not a short-term decision, so be prepared to be breaking bread with your vendor for some time to come.

IV. Readiness of Product

XI benefits from coming late to the market. It has had an opportunity to learn from the pioneering technology approaches of its competitors. Rather than relying on older techniques of proprietary scripting or ad hoc user-created structures, SAP has embraced the Web services model for all communication internally within XI, using SOAP, HTTP, and XML conventions. Where the competition may execute in the “plain” Java environment, SAP has embraced full-blown J2EE as its execution environment. Moreover, SAP complies with at least as many standards out of the box as do its competitors, including the use of the Java Connector Architecture to enable lightweight adapters.

Also by coming late, SAP has been able to avoid some of the over-engineered approaches (although not without a few of its own; try to get any XI developer to explain the added value of the System Landscape Directory) of the past, such as the bullet-proof, yet burdensome use of CORBA and internal MQ within the Websphere Interchange Server. This is also true of its good-intentioned (but only occasionally leveraged) approach of sometimes redundant secondary mappings to canonical data objects.

Very notably, XI has been developed by a single vendor as a unified offering, so SAP has not had to cobble different run-time engines for B2B, EAI, and BPM, covered over by a thin veneer of common development tools. Five years after the ActiveWorks acquisition, webMethods still runs a separate EAI broker to enable guaranteed pub/sub integrations. Although perhaps not without its own generous helping of extra layers of abstraction, a clean design slate has enabled SAP to deliver a coherence of functionality resulting from a common design. This is not a small thing from a development and support standpoint, something an evaluator should keep in mind.

A. Performance/Reliability
SAP's approach of using ABAP for the main integration engine, and J2EE for everything else (development tools, adapter, mapping, and the SLD) has been criticized from all sides. From a customer's architectural and administration standpoint, it requires skill sets comfortable in both camps, which is still a scarce and expensive commodity. The resulting complexity smacks of a committee compromise.

On the other hand, there is no way J2EE threads can today deliver the bulletproof transaction security (and hence guaranteed integration message atomicity) of Basis work processes, a capability depended upon by thousands of businesses everyday. Moreover, there is no way J2EE today can demonstrate the scalability of the ABAP environment, the latter benchmarked to stratospheric heights. SAP's approach of leveraging the proven ABAP runtime will, in retrospect, prove to have been a wise move.

B. Publish/Subscribe

The widespread metaphor of a sender anonymously making available a business document (publish), and an anonymous receiver choosing to receive it (subscribe) without concern as to the sender's identity, is not really addressed in XI. Consequently, XI has been accused of being an inherent, point-to-point architecture, exhibiting little of the publish/subscribe model. Ignoring, for the moment, the dirty secret that arguably a majority of EAI scenarios anywhere *are* point-to-point (with a bothersome percentage just shuffling flat files) – this is a fair observation. Instead of delivering the package to "Occupant", valid receivers (even if wildcarded) for a given sender and object type are defined at configuration time (but not at development time). Although simple enough to change this configuration, SAP seems to impose a priori control over who exactly may receive a given document from a given sender, and under what conditions. Deterministic (if predefined) message routing may be more important than anonymous pub/sub in any integrations involving Sarbanes-Oxley compliance.

C. Synchronous / Asynchronous

XI strongly enforces either a synchronous or asynchronous collaboration across an entire given scenario between sender and receiver. It's not possible to feature, say a synchronous sender and an asynchronous receiver, without resorting to BPM. Rigorous adherence to the model may cramp the way companies do business. For example, the purchase order create scenario from SRM with an HTTP post to the supplier's secure Website is an asynchronous model – but the business may want to check the always-present HTTP response to ensure a good return code (200), and do something with that return code. XI users are out of luck if they want to examine that return code in such an async scenario.

D. Development Tools

I've mainly discussed runtime environments and not development tools. This is quite opposed to the traditional approach for evaluating competing integration software offerings, which usually zooms in quickly on the development tools in general, and mapping tools in particular. Although stunning in the integration product swimsuit competition, mapping tools provide structure and value transformations; hopefully with a library of supporting services (vital functions, but only part of the bigger picture).

Experienced integration developers will tell you that half their time is spent working with users, anticipating unspoken needs, and using experience to fill undefined gaps. The other half of their time is spent testing. The third half? Or, as some would say, where do you get the time for development? With competent specifications, actual development using the GUI tools takes up a relatively small proportion of time. None of this is to say that graphical mapping tools are unimportant, but rather that most vendors deliver satisfactory development tools.

As a developer, I can usually understand what my solution does through the natural progression of the development process. What separates tools from each other is how easy it is for the developer coming in *next* year, to figure out exactly what I did today. Because of an integration product's bias toward decoupling components for later possible reuse, it is not always obvious in the graphical tool how they are strung together to deliver an actual end-to-end scenario.

This may mean that the developer who comes around next year spends a great deal of time crawling around under the floorboards, figuring out which cable goes where in my scenario. Naming conventions help here, as does good documentation practice. At first glance, it appears that XI joins right in the confusion with the Integration Repository/ Design Time tool, whose objects individually provide only a portion of a scenario. However, XI does provide a good means for apprehending the entire flow with a pair of tools: the Integration Repository's graphical business scenarios, and the Integration Directory/Configuration Time optional business scenarios. (Lest this make things *too* easy, an XI Configuration Time business scenario may or may not have anything to do with a Design Time business scenario.)

E. XI's Knowledge of SAP Business Applications
SAP makes a strong claim that XI has the advantage of knowing about all SAP applications, and can consequently shortcut development by importing the parameter signature and structure of any RFC, BAPI, or IDoc. True, but the other vendors do this as well, since SAP has graciously provided common RFCs to enable this (for example, RFC_GET_FUNCTION_INTERFACE, RFC_GET_STRUCTURE_DEFINITION).

F. Business Process Modeling/Monitoring (BPM)
Just about all integration vendors now provide a means to enable long-lived transactions, ones featuring logic processing with time or sequence-based dependencies on external business events. Prior to this capability, this sort of master business process monitoring and correlation logic had to be built into the application side, for example, in custom ABAP. Much of that sort of intelligence can now be rendered in integration vendors' BPM engines (whether it *should* be is a matter for a different discussion, as some customers choose specifically to keep business logic close to its source/target, rather than on a high-stepping integration engine optimized for rapidly trading data in the pits). And just as the industry begins to recognize the value of a BPM mechanism, it leapfrogs itself and rushes toward the distant horizon of Composite Applications, relegating mere BPM to a more modest, though fully useful, career (another topic for future discussion).

XI itself has been optimized to move messages efficiently, using its BPM to perform tasks that competing products can sometimes do as a matter of course within the integration engine itself (e.g., correlation of multiple dependent message inputs before triggering a downstream message). Fortunately, SAP's BPM is built upon the tried and true SAP Business Workflow engine, although it does not depend upon SAP's typical workflow tools.

G. Adapters
In the early days of integration products, vendors cataloged a suite of intelligent adapters that leveraged the idiosyncrasies of specific applications, and in some cases even for specific business processes within those applications. That a vendor could offer adapters to simultaneously support multiple versions of Baan and Clarify, Oracle Financials *and* Oracle database, was perceived as a competitive advantage. However, the ongoing support and development to offer a broad range of adapters in anticipation of hoped-for demand proved costly, and vendors backed off the numbers game.

Today, even database vendor-specific and queuing product-specific adapters are being deprecated by vendors in favor of more generic, standards-driven, technical adapters, (such as JDBC and JMS). SAP itself offers only SAP application (ALE, RFC) and technical (e.g., non-application specific) adapters, relying, as does Microsoft, on specialized third-party vendors to fill the gaps. iWay and SeeBurger, for example, provide certified XI PeopleSoft or EDI adapters.

While this may be a valuable division of labor, it also presents a potential risk, since the success of the XI implementation may depend on support from a comparatively small vendor. For SAP customers choosing XI specifically to avoid vendor finger pointing, in-depth customer reference analysis of anticipated third-party adapters is recommended.

H. Business Activity Monitoring (BAM)
Vendors are incorporating a BAM layer on their integration tools, with real-time thresholds set to monitor related statuses across a number of disparate applications. While SAP offers building block products that could enable a BAM solution (e.g., Business Warehouse, XI, and Event Management), this additional reporting layer is today a gap in the XI product set.

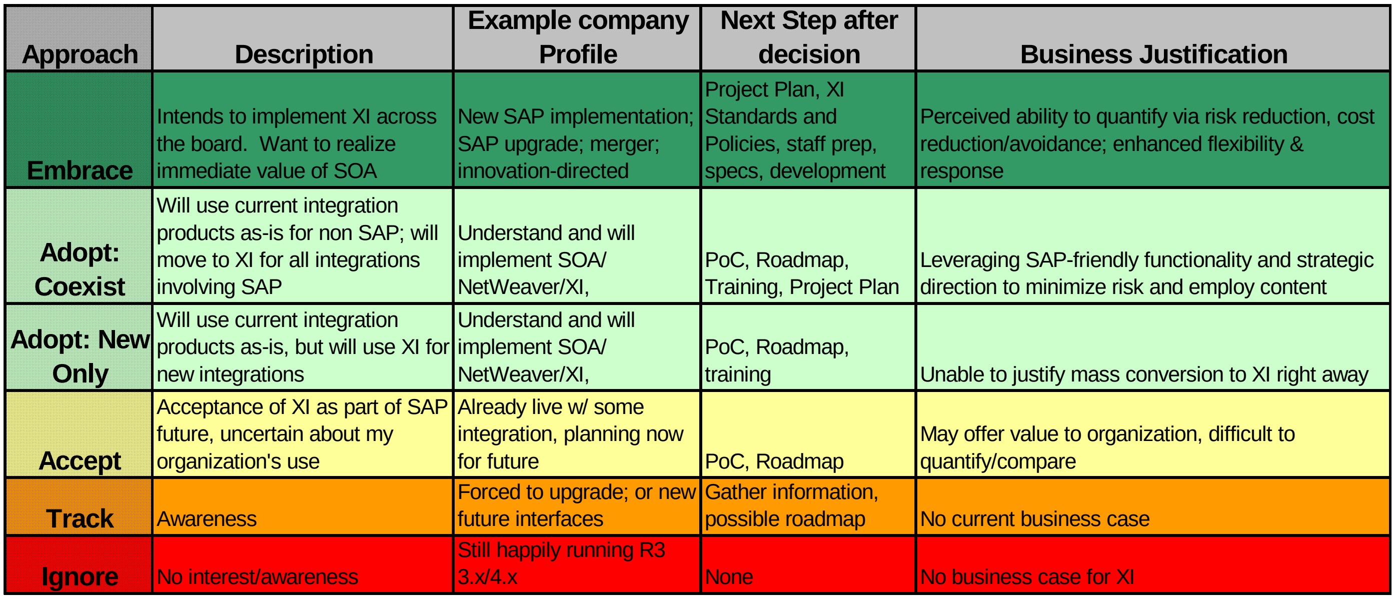
I. Extract-Transform-Load (ETL)
There are a few good ETL tools available that can generate ABAP to provide mass transports of large amounts of data on a regular basis. These are used widely for synchronization across data stores within an organization. Although certainly technically feasible, this is a specialty that EAI and B2B tools by and large don't address. XI is not an ETL product.

J. Master Data and Common Canonical Business Objects
Many organizations have taken the challenge of commonizing data object types with varying success. While the rewards may be significant, the task has often proved greater than anticipated. With the introduction of integration products, many customers anticipated the possibility of defining a universal representation of, say a customer master record, that could then be transformed by the middleware into this or that application's own peculiar notion of that object. Things didn't really work out that way, whether because of product limitations or organizational ones. SAP does provide the Master Data Management (MDM) product to address this; a more functional approach than trying to do so through the middleware (although MDM does depend on XI for reliable data transport).

K. Upgrades
Ask vendors what their product upgrade strategy is, and what they do to support it. Hopefully there is a smooth path. With luck, a complete migration to a fresh installation of the new version is not necessary. Also ask vendors what their history is for providing upgrade tools and utilities, not just for the initial upgraded release, but tools maintained as long as the old release is in general support. An upgrade of a complex environment three years down the road can cost nearly as much in effort and currency as the original implementation.

L. Maturity
With the release of 3.0, XI became a viable player in the integration space. The product works well, and with a reliability that properly reflects the pure necessity of reliability in a guaranteed delivery product. Most of the more recent XI support packages have continued to feature new functionality. This proved a way for SAP to quietly introduce it in incremental fashion, given the cancellation of the full-blown NetWeaver 05 release. As a core product, XI can stand today with the best of its brethren, although in some niche or specialty features, it may fall short. This is not unexpected from a product rapidly expanding its feature set, but it underscores the importance of clearly defining specifically what is required from the product during the evaluation.

M. Methodology
Some vendors publish a nicely packaged implementation methodology, although content may sometimes be sparse. Such lightness may be deliberate, relying on vendor professional services to really fill things out, or it may simply be a reflection that no two implementations are the same, and that methodologies provide frameworks for an organic process, rather than a canned dance pattern. SAP's methodology (nee ASAP, now Solution Manager) content for XI is available from SDN to SAP customers. Unfortunately, it is extremely sparse. PowerPoints from previous TechEds may provide good product information perhaps, but may not be implementation specific. Other offerings, such as the webMethods GEAR methodology, are head and shoulders above this.



Approach	Description	Example company Profile	Next Step after decision	Business Justification
Embrace	Intends to implement XI across the board. Want to realize immediate value of SOA	New SAP implementation; SAP upgrade; merger; innovation-directed	Project Plan, XI Standards and Policies, staff prep, specs, development	Perceived ability to quantify via risk reduction, cost reduction/avoidance; enhanced flexibility & response
Adopt: Coexist	Will use current integration products as-is for non SAP; will move to XI for all integrations involving SAP	Understand and will implement SOA/ NetWeaver/XI,	PoC, Roadmap, Training, Project Plan	Leveraging SAP-friendly functionality and strategic direction to minimize risk and employ content
Adopt: New Only	Will use current integration products as-is, but will use XI for new integrations	Understand and will implement SOA/ NetWeaver/XI,	PoC, Roadmap, training	Unable to justify mass conversion to XI right away
Accept	Acceptance of XI as part of SAP future, uncertain about my organization's use	Already live w/ some integration, planning now for future	PoC, Roadmap	May offer value to organization, difficult to quantify/compare
Track	Awareness	Forced to upgrade; or new future interfaces	Gather information, possible roadmap	No current business case
Ignore	No interest/awareness	Still happily running R3 3.x/4.x	None	No business case for XI

Figure 3: XI Customer Adoption Patterns

V. XI Adoption Approaches

Over the last few years, I've had conversations with many clients about their position with regard to ESA, NetWeaver, and XI. Whether or not XI is on their immediate horizon, most SAP customers have at least an interest in, and recognize the advisability of, movement toward XI. A surprisingly high percentage of them accept the inevitability of XI (Figure 3).

For SAP customers already live with third-party integration products, there would at first glance appear to be little incentive to consider a move. But annual maintenance charges for the third-party software, coupled with the allure of SOA and the built-in coherence between SAP apps and XI, may provide a business case for XI. Even when there may be no immediate justification for a replacement strategy, customers facing a major upgrade of their existing integration product have an opportunity to consider XI as an alternative (as do those facing a major business event, such as an acquisition or merger).

A key tool for some customers is the XI integration project roadmap. When delivering these tailored deployment strategies for clients I've analyzed, current and future integration needs the best roll-in strategy for XI, organizational readiness, and business justification. An important part of this is the prioritization and phasing of scenarios offering the greatest immediate return at the most modest risk.

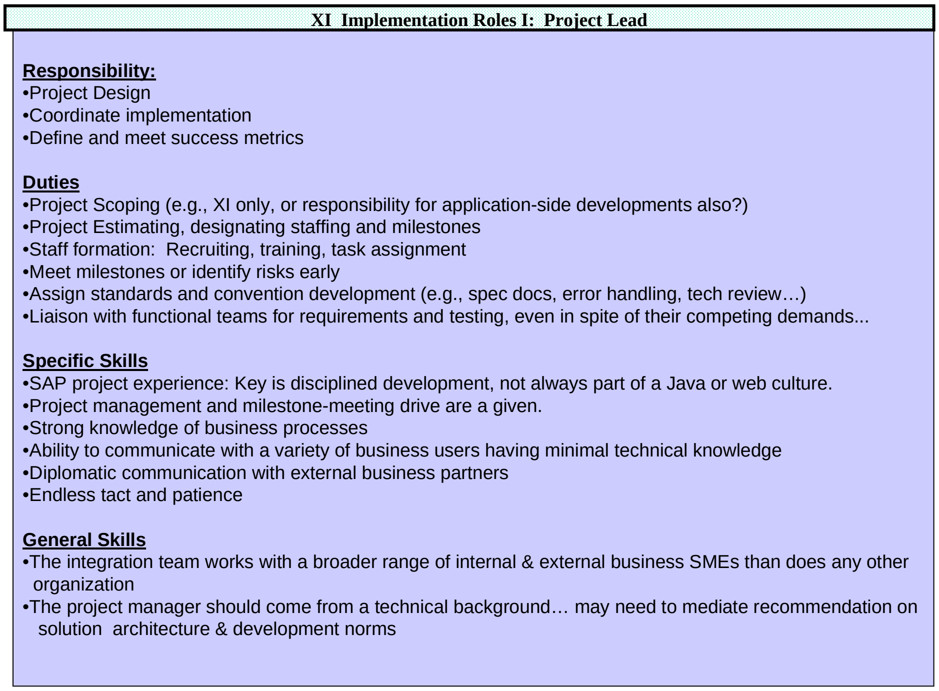
A second tool is the proof of concept (PoC): simply defining a representative subset of existing interfaces, and deploying them on an XI system. This gives a very good indication of what is involved, and more importantly certifies the feasibility of XI.

For new SAP customers, or those first implementing an integration project, XI may well be a natural choice. For such customers, I've moved more directly to the XI project planning and implementation tasks. PoCs may still be a valuable tool, first, to validate suitability, but also to begin to build a reference set of scenarios that developers can reuse and adapt during the implementation phase.

VI. Readiness of the Organization

XI is not necessarily an intuitive environment to learn, even for those having either EAI / B2B product experience, or traditional SAP R/3 integration experience. Once learned, the tools work quite well, and the developer can be very productive, but there is a definite learning curve.

Specific roles of the XI team are not different from those for most development. But specific duties certainly differ from those in the traditional ABAP environment (Figures 4, 5, 6).



XI Implementation Roles I: Project Lead

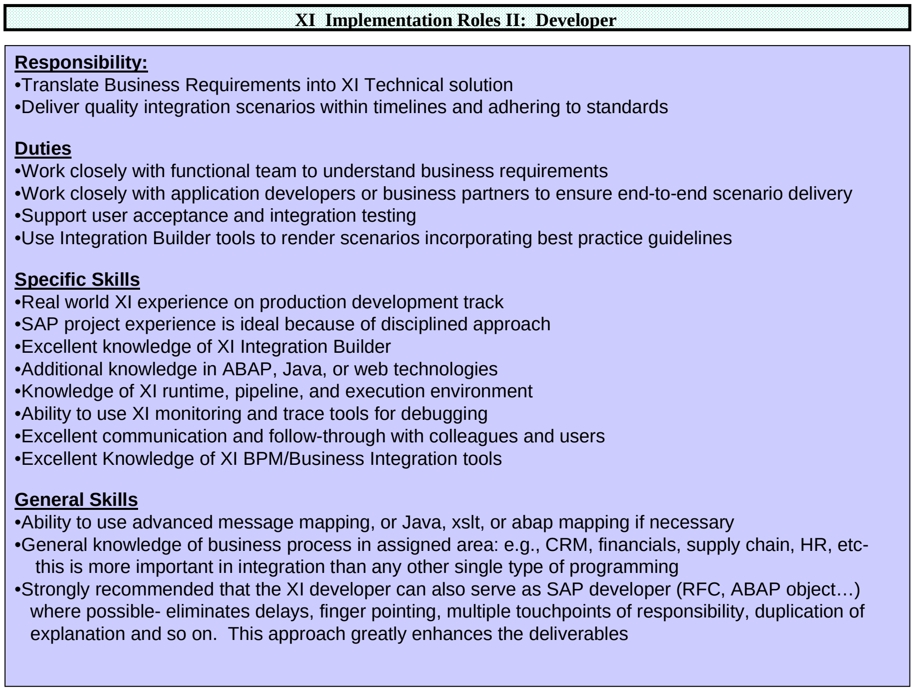
Responsibility:
- Project Design
- Coordinate implementation
- Define and meet success metrics

Duties
- Project Scoping (e.g., XI only, or responsibility for application-side developments also?)
- Project Estimating, designating staffing and milestones
- Staff formation: Recruiting, training, task assignment
- Meet milestones or identify risks early
- Assign standards and convention development (e.g., spec docs, error handling, tech review…)
- Liaison with functional teams for requirements and testing, even in spite of their competing demands...

Specific Skills
- SAP project experience: Key is disciplined development, not always part of a Java or web culture.
- Project management and milestone-meeting drive are a given.
- Strong knowledge of business processes
- Ability to communicate with a variety of business users having minimal technical knowledge
- Diplomatic communication with external business partners
- Endless tact and patience

General Skills
- The integration team works with a broader range of internal & external business SMEs than does any other organization
- The project manager should come from a technical background… may need to mediate recommendation on solution architecture & development norms

Figure 4: XI Project Lead Description



XI Implementation Roles II: Developer

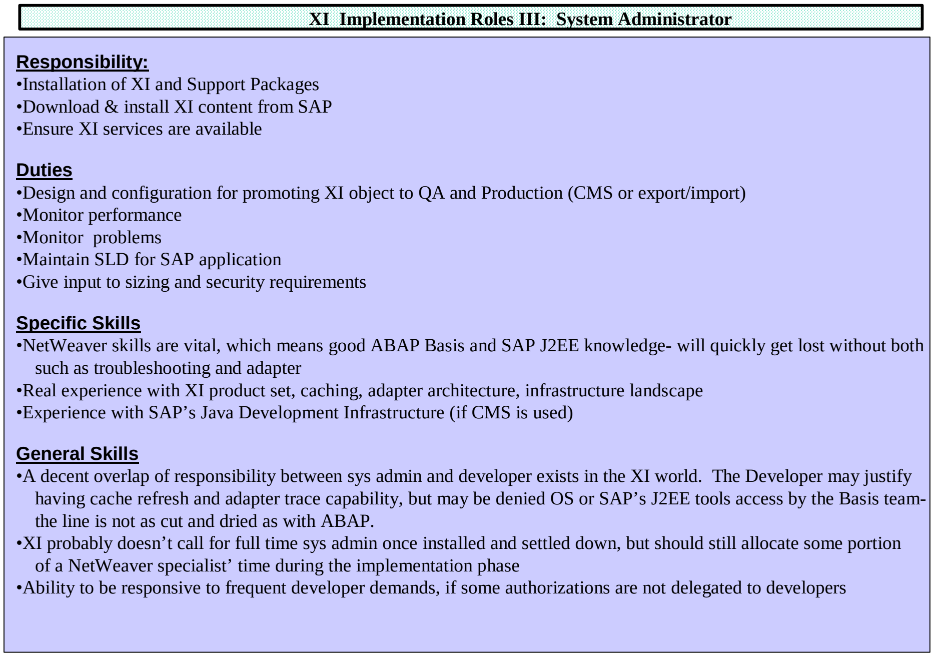
Responsibility:
- Translate Business Requirements into XI Technical solution
- Deliver quality integration scenarios within timelines and adhering to standards

Duties
- Work closely with functional team to understand business requirements
- Work closely with application developers or business partners to ensure end-to-end scenario delivery
- Support user acceptance and integration testing
- Use Integration Builder tools to render scenarios incorporating best practice guidelines

Specific Skills
- Real world XI experience on production development track
- SAP project experience is ideal because of disciplined approach
- Excellent knowledge of XI Integration Builder
- Additional knowledge in ABAP, Java, or web technologies
- Knowledge of XI runtime, pipeline, and execution environment
- Ability to use XI monitoring and trace tools for debugging
- Excellent communication and follow-through with colleagues and users
- Excellent Knowledge of XI BPM/Business Integration tools

General Skills
- Ability to use advanced message mapping, or Java, xslt, or abap mapping if necessary
- General knowledge of business process in assigned area: e.g., CRM, financials, supply chain, HR, etc- this is more important in integration than any other single type of programming
- Strongly recommended that the XI developer can also serve as SAP developer (RFC, ABAP object…) where possible- eliminates delays, finger pointing, multiple touchpoints of responsibility, duplication of explanation and so on. This approach greatly enhances the deliverables

Figure 5: XI Developer Description



XI Implementation Roles III: System Administrator

Responsibility:
- Installation of XI and Support Packages
- Download & install XI content from SAP
- Ensure XI services are available

Duties
- Design and configuration for promoting XI object to QA and Production (CMS or export/import)
- Monitor performance
- Monitor problems
- Maintain SLD for SAP application
- Give input to sizing and security requirements

Specific Skills
- NetWeaver skills are vital, which means good ABAP Basis and SAP J2EE knowledge- will quickly get lost without both such as troubleshooting and adapter
- Real experience with XI product set, caching, adapter architecture, infrastructure landscape
- Experience with SAP's Java Development Infrastructure (if CMS is used)

General Skills
- A decent overlap of responsibility between sys admin and developer exists in the XI world. The Developer may justify having cache refresh and adapter trace capability, but may be denied OS or SAP's J2EE tools access by the Basis team- the line is not as cut and dried as with ABAP.
- XI probably doesn't call for full time sys admin once installed and settled down, but should still allocate some portion of a NetWeaver specialist' time during the implementation phase
- Ability to be responsive to frequent developer demands, if some authorizations are not delegated to developers

Figure 6: XI System Administrator Description

A. Consulting
At least for the present, organizations implementing XI will often benefit by tapping the experience of a seasoned XI consultant. While skills such as SAP-FI or ABAP may often be picked up from colleagues (or as a matter of course during an implementation), a widespread body of XI knowledge is just not yet available. SAP's admirable and freely joinable SAP Developers Network (SDN) may feature very good information, or, in the tradition of the Internet, frustratingly spotty responses to complex questions, with the toughest questions too often unanswered. Integration is no place to fool around: a client should insist on diverse real-world XI experience in particular, and integration experience in general. Further, a client should look for experience in production track integrations, not just those rendered as part of a pilot project. Lists of sample questions for the XI certification exams have been circulated, so it probably goes without saying that simply training or certification are themselves probably not sufficient proficiency indicators.

B. Organization
Integration development and production support don't always fit well into an IT organizational structure. Lines between system admin and development blur; developers may have more experience in the legacy app than SAP or vice versa. They may have more experience with Java and Web interfaces than in ABAP Object troubleshooting. Is XI supported by the ABAP team, or the CICS team, or the Java team? Organizations of any size, sometimes deliberately and sometimes by accident, move to a cross-disciplinary approach of an integration Competency Center. This may comprise individuals who are XI experts, or those having technical experience in the applications to communicate via XI. Some of the members can be conversant with Web services and Java, while others may know the homegrown app and the supplier and banker networks with whom we converse. This team is also responsible for developing sample reference integrations, standards, and procedures. From a support perspective, the Competency Center can provide specialized support right behind the help desk. If need be, they can additionally triage tickets to the specialized application or technical area (another topic for a follow on paper).

VII. Conclusion

Client server computing is not even a point/counterpoint topic today; it is mainstream. No more trade journal articles debate its merits, no trade show presentations relate cases demonstrating pure client-server computing, and no industry analysts provide surveys predicting its adoption. But prior to R/3 Basis, it was all but impossible to find client-server implementations that had more than a few dozen users. In effect, SAP R/3 legitimized enterprise-wide client-server computing. And SAP is poised to do the same thing to the SOA model with its Enterprise Services Architecture (ESA). Other vendors are offering SOA tools, platforms, and environments, but SAP is stealing a march on its integration rivals by baking its own applications directly into its tools, and so providing an argument for SOA that is justifiable to the business. No other vendor is so well positioned to do so.

SAP is going beyond mere branding of an industry buzzword. In effect SAP is betting its future growth and viability upon the ESA model. Beyond the millions in development funds, the application architecture is built upon the premise of the ESA. If the bet is successful, SAP extends its dominance for another decade; if not, the smaller or more specialized ERP competitors begin rapidly catching up.

The integration problem is valid: integration has not been getting easier – but then the target is moving also. Let me restate, pure integration is indeed easier today, and measurable benefit is available just from an integration hub. Once the hub is in place, hosting and publishing cataloged services is indeed simplified. And once we are in such an architected position, integration actually *does* get easier; hopefully at much more benefit to the business.

XI is a fundamental enabler of the ESA vision, and the fate of ESA is intimately tied to it. No doubt XI can serve well enough on its own as simple EAI and B2B tools, and this will be sufficient for a large number of companies. However, the beach-head has already been established. It's no longer a good bet that XI will survive. The action has passed into the question of how fast it will spread inland.

Dave Bernard, KimberWorks, Inc.**,** *has been kicking around the SAP world for a dozen years, and the EAI world for six. In addition to stints as an engineer for Sun and HP, he was an early employee of CrossWorlds (now IBM Websphere), and has since implemented multiple complex WBI and webMethods projects. With the announcement of NetWeaver, SAP recruited Dave to help develop its fledgling XI consulting practice, where he put together some of the original XI estimating guidelines, implementation roadmaps, and methodologies. In addition to working as a NetWeaver project manager, and as an XI "Meet the Expert" at tradeshows such as ASUG, he delivered numerous NetWeaver/ESA/XI roadmaps and customer workshops. Today Dave is continuing to serve clients as a NetWeaver and XI architect and hands-on developer, and is making a point never to use the word "ecosystem." Dave can be reached at* dave.bernard@saptips.com.

NO WARRANTY: This documentation is delivered as is, and Klee Associates, Inc. makes no warranty as to its accuracy or use. Any use of this documentation is at the risk of the user. Although we make every good faith effort to ensure accuracy, this document may include technical or other inaccuracies or typographical errors. Klee Associates, Inc. reserves the right to make changes without prior notice.

NO AFFILIATION: Klee Associates, Inc. and this publication are not affiliated with or endorsed by SAP AG. SAP AG software referenced on this site is furnished under license agreements between SAP AG and its customers and can be used only within the terms of such agreements. SAP AG and mySAP are registered trademarks of SAP AG.

All other company and product names used herein may be trademarks or registered trademarks of their respective owners.